WE CLAIM:

1. A method for multicasting data messages to members of a multicast group, the multicast group comprising a sequencer, one or more clients, one or more data servers, and one or more commit servers, the method comprising the steps of:

transmitting a first data message to the members of the multicast group;

each data server that receives the first data message requesting the sequencer to assign a first sequence number to the first data message, the first sequence number being from a sequence of numbers allocated to the data messages, said first sequence number following all sequence numbers assigned prior to assignment of the first sequence number;

assigning the first sequence number to the first data message, in response to the sequencer receiving a first quantity of the requests to assign a first sequence number to the first data message;

notifying the commit servers of the assignment of the first sequence number to the first data message;

each of the commit servers sending to the sequencer an acknowledgment of the notification of the assignment of the first sequence number to the first data message, in response to being notified of the assignment of the first sequence number to the first data message;

committing the first sequence number to the first data message, in response to the sequencer receiving a second quantity of the acknowledgments of the notification of the assignment of the first sequence number to the first data message; and

informing the members of the multicast group of the commitment of the first sequence number to the first data message.

2. A method according to Claim 1, wherein

said step of each data server that receives the first data message requesting the sequencer to assign a first sequence number to the first data message includes the step of sending, from said each data server that receives the first data message to the sequencer, a data report message identifying the first data message;

said step of notifying the commit servers of the assignment of the first sequence number includes the step of submitting to the commit servers a commit submit message identifying the first data message;

said step of sending to the sequencer an acknowledgment of the notification of the assignment of the first sequence number includes the step of sending to the sequencer a commit acknowledge message identifying the first data message; and

said step of informing the members of the multicast group of the commitment of the first sequence number includes the step of sending a commit message identifying the first data message to the members of the multicast group.

3. A method according to Claim 2,

further comprising the step of transmitting a second data message to the members of the multicast group;

wherein said step of sending, from said each data server that receives the first data message to the sequencer, a data report message identifying the first data message further includes the step of a first data server sending a first data report message identifying the first data message to the sequencer after said first data server receives the second data message, said first data report message also identifying the second data message.

4. A method according to Claim 2, further comprising the steps of:

transmitting a second data message to the members of the multicast group;

each data server that receives the second data message requesting the sequencer to assign a second sequence number, the second sequence number being

from the sequence of numbers allocated to the data messages, said second sequence number following all sequence numbers assigned prior to assignment of the second sequence number, to the second data message, said step of each data server that receives the second data message requesting the sequencer to assign a second sequence number to the second data message, includes the step of sending from said each data server that receives the second data message to the sequencer a data report message identifying the second data message;

assigning the second sequence number to the second data message, in response to the sequencer receiving a third quantity of the requests to assign a second sequence number to the second data message;

wherein said step of notifying the commit servers of the assignment of the first sequence number further includes the step of notifying the commit servers of the assignment of the second sequence number, said commit submit message identifying the first data message also identifying the second data message.

5. A method according to Claim 2, wherein the members of the multicast group deliver the data messages to their respective upper layer applications in order of progressing sequence numbers, further including the step of using a receiver driven, negative acknowledgment-based approach to improve reliability of delivery of the data messages.

6. A method as in any one of Claims 1-5, wherein said data servers store said data messages transmitted to the multicast group, the multicast group further comprising checkpoint servers, the method further including the steps of:

step for message consolidation;

step for garbage collection; and

step for storing said first sequence number in stable storage.

7. A method for processing data messages multicast to members of a multicast group, the multicast group comprising a sequencer, one or more clients, one or more data servers, and one or more commit servers, the method comprising the steps of:

each data server that receives said each data message requesting the sequencer to assign a sequence number, from a sequence of numbers allocated to the data messages, to said each data message, in response to receiving each data message;

assigning a sequence number following all sequence numbers assigned prior to assignment of the sequence number to said each data message, in response to the sequencer receiving a first quantity of requests to assign a sequence number to said each data message;

notifying the commit servers of each assignment, each notification identifying said each assignment by said each data message and the sequence number assigned to said each data message;

each of the commit servers sending to the sequencer an acknowledgment of said each notification, in response to being notified of said each assignment, said acknowledgment identifying said each data message;

committing said each assignment, in response to the sequencer receiving a second quantity of the acknowledgments identifying said each data message; and

informing the members of the multicast group of each commitment.

8. A method according to Claim 7, wherein

the members of the multicast group deliver the data messages to their respective upper layer applications in order of progressing sequence numbers;

said data servers store said data messages transmitted to the multicast group;

further including the step of using a receiver driven, negative acknowledgment-based approach to improve reliability of delivery of the data messages.

9. A method according to Claim 8, wherein said each data message is associated with a unique message ID and is identifiable from its associated message ID, the step of using further includes the steps of:

each member of the multicast group identifying gaps in a progression of sequence numbers known by said each member of the multicast group to have been committed to data messages received by said each member of the multicast group;

if said each member of the multicast group does not know a first message ID, said first message ID being associated with a first data message, a first sequence number within one of said gaps having been previously committed to said first data message, said each member of the multicast group querying one of said commit servers to obtain said first message ID; and

if said each member of the multicast group has not received said first data message, querying one of said data servers to retrieve said first data message.

10. A method according to Claim 8, wherein said each data message is associated with a unique message ID and is identifiable from its associated message ID, the step of using further includes the steps of:

each member of the multicast group identifying gaps in a progression of sequence numbers known by said each member of the multicast group to have been committed to data messages received by said each member of the multicast group;

said each member of the multicast group querying one of said data servers to retrieve said first data message.

11. A method according to Claim 10, further comprising the step of said sequencer periodically generating and sending heartbeat messages to the members of the multicast group, each said heartbeat message containing an associated largest sequence number, said associated largest sequence number being the last sequence number committed at a time substantially equal to a time said heartbeat message is generated.

12. A method according to Claim 8, further comprising the step for periodic message consolidation.

13. A method according to Claim 8, wherein the multicast group further comprises one or more checkpoint servers, the method further comprising the step of performing periodic message consolidation by said checkpoint servers at message intervals determined through a common consensus protocol, each message consolidation producing a checkpoint associated with said each message consolidation, said checkpoint associated with said each message consolidation corresponding to a terminal data message, said checkpoint associated with said each message consolidation containing checkpoint information, the checkpoint information being sufficient for a first upper layer application of said upper layer applications to reconstruct a cumulative system state said first upper layer application would attain upon receiving said terminal message and all said data messages that preceded said terminal message.

14. A method according to Claim 13, further comprising the step of said checkpoint servers periodically generating and sending checkpoint reports to said sequencer, each checkpoint report corresponding to latest checkpoint at the time said each checkpoint report is generated, said each Checkpoint Report identifying a sequence number of its corresponding terminal data message, said each checkpoint report carrying size data of the latest checkpoint.

15. A method according to Claim 14, further comprising step for synchronizing a first asynchronous upper layer process of a first asynchronous member of the multicast group with other members of the multicast group, said first asynchronous member not being said sequencer or one of said data or commit servers.

16. A method according to Claim 14, further comprising the step of synchronizing a first asynchronous upper layer process of a first asynchronous member of the multicast group with other members of the multicast group, said first asynchronous member not being said sequencer or one of said data or commit servers, said synchronizing step including the steps of:

said first asynchronous member retrieving a first checkpoint from said checkpoint servers;

Rule 1.126 17

18. A method according to Claim 16, wherein said each data message bears a corresponding logical timestamp, said logical timestamp including a most recent sequence number known to original sender of said each data message when said each data message was first sent.

18

19. A method according to Claim 18, further comprising the step of:

the data servers deleting said stored messages that have logical checkpoints older by a maximum logical lifetime number at the time of deletion than a most recent sequence number known at the time of deletion.

19

20. A method according to Claim 14, further comprising the step of:

said data servers deleting the stored data messages that are older than the latest checkpoint.

20

21. A method according to Claim 16, wherein the multicast group further includes stable storage writeable by said sequencer, said method further comprising the step of said sequencer storing in said stable storage said assigned sequence number before said step of notifying the commit servers.

22. A method according to Claim 8, wherein the multicast group further includes stable storage writeable by said sequencer, said method further comprising the step of said sequencer storing in said stable storage said assigned sequence number before said step of notifying the commit servers.